Министерство науки и высшего образования РФ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Глазовский государственный педагогический институт имени В.Г. Короленко»

**ОТЧЕТ**

**по учебной практике**

**ПМ 01 Разработка программных модулей программного обеспечения для компьютерных систем**

**Специальность 09.02.03 Программирование в компьютерных системах**

|  |
| --- |
| Выполнил: |
| студент ОП СПО группа 33 |
| Третьяков А.В. |
|  |
| Руководитель практики: старший преподаватель кафедры математики и информатики Касаткин К.А. |

г. Глазов 2020

**Оглавление**

1. [**Техническое задание на создание программного обеспечения**](#_5e8jemgjm1f)
2. [**Актуальность**](#_ky3gbotk8zf9)
3. [**Аналоги ПО**](#_j6zpq938hapj)
4. [**Используемые технические и программные средства**](#_8i370of7b9yy)
5. [**Формализация бизнес-процессов предметной области**](#_jfpb90kzhf2v)
6. [**Описание программы**](#_wzawwu514fmm)
7. [**Исходный код**](#_xc23f5jwz7s8)
8. [**Отчет по тестированию**](#_rkfk9dmwil3a)
9. [**Руководство пользователя**](#_1s9i6edd4q59)
10. [**Руководство системного администратора**](#_9xub4386gzz5)
11. [**Список источников**](#_otfqcn8xahlj)

### **Техническое задание на создание программного обеспечения**

1. **Цель**: написать программу, которая будет определять цвет пикселя на изображении
2. **Основные задачи:**

2.1 Определять цвет пикселя

2.2 Загрузка фотографии

1. **Используемые программные и технические средства:**

3.1 Персональный компьютер стандартной конфигурации

3.2 Операционная система windows 10

3.3 Средство разработки QtCreator v4.11

3.4 Язык программирования C++

1. **Требования к системе:**

4.1 Определять цвета пикселей

4.2 Правильно распознавать цвета и оттенки

4.3Возможность загрузки своего изображения

1. **Состав документации:**

5.1Руководство пользователя

5.2 Руководство системного администратора

### Актуальность

Нас постоянно окружают тысячи цветов и десятки тысяч оттенков. У многих людей часто возникают задачи, связанные с подбором правильного оттенка. Это дизайнеры, контент-менеджеры, программисты и многие другие. Для облегчения их работы было создано данное приложение. Данное приложение позволяет узнать не только цвет пикселя, но и к тому же его код. Это очень удобно при работе с различными оттенками.

### Аналоги ПО

**Ecolor**

EColor — это утилита для мгновенного определения цвета пикселя на экране компьютера. Данная утилита находится в режиме ожидания в системном лотке до появления события нажатия запрограммированного сочетания клавиш на клавиатуре. При возникновении данного события утилита определяет цвет пикселя в месте положения курсора мыши.

**Преимущества:**

1. EColor выдаёт информацию о коде цвета пикселя в следующих форматах:

* HEX
* RGB
* CMYK
* HSV
* HSL

1. В настройках приложения можно задать комбинацию клавиш, при которых срабатывает захват цвета пикселя под курсором, а также установить автозапуск приложения при старте операционной системы. В настройках приложения имеется возможность включить автоматическое копирование кода цвета в буфер обмена.
2. Утилита имеет функционал лупы для удобства захвата целевого пикселя, а также возможность задания координат целевого пикселя.
3. Уведомление с кодом цвета пикселя может отображаться как в правом нижнем углу экрана, так и рядом с курсором мыши. Имеется возможность перетаскивать окно утилиты по рабочему столу.

**Недостатки:**

Не обнаружены

**Сolourco**

Интересный и очень необычный color picker, который представлен в виде web-страницы, меняющей свой цвет в зависимости от позиции курсора. Также на сайте представлены различные палитры и цветные схемы, работающие по схожему принципу. Когда пользователь находит цвет который ему понравился, для дальнейшей работы нужно, всего лишь, нажать на код цвета. После чего мы сможем изменить оттенок, а также увидеть его запись в разных форматах.

**Преимущества:**

1. Возможность найти нужный оттенок
2. Присутствует код оттенка/цвета

**Недостатки:**

1. Отсутствует возможность загрузки своего изображения
2. Невозможно определить цвет пикселя, только самостоятельный подбор
3. Цвет показывается только в одних форматах

**Color picker (от google)**

Для того чтобы получить простенький колор пикер, нужно, всего лишь, ввести в гугл ключевой запрос «color picker», после этого вам отобразится эта палитра.

**Преимущества:**

1. Выдает информацию о коде цвета пикселя в следующих форматах:

* HEX
* RGB
* CMYK
* HSV
* HSL

1. Можно найти точный цвет/оттенок, который нужен

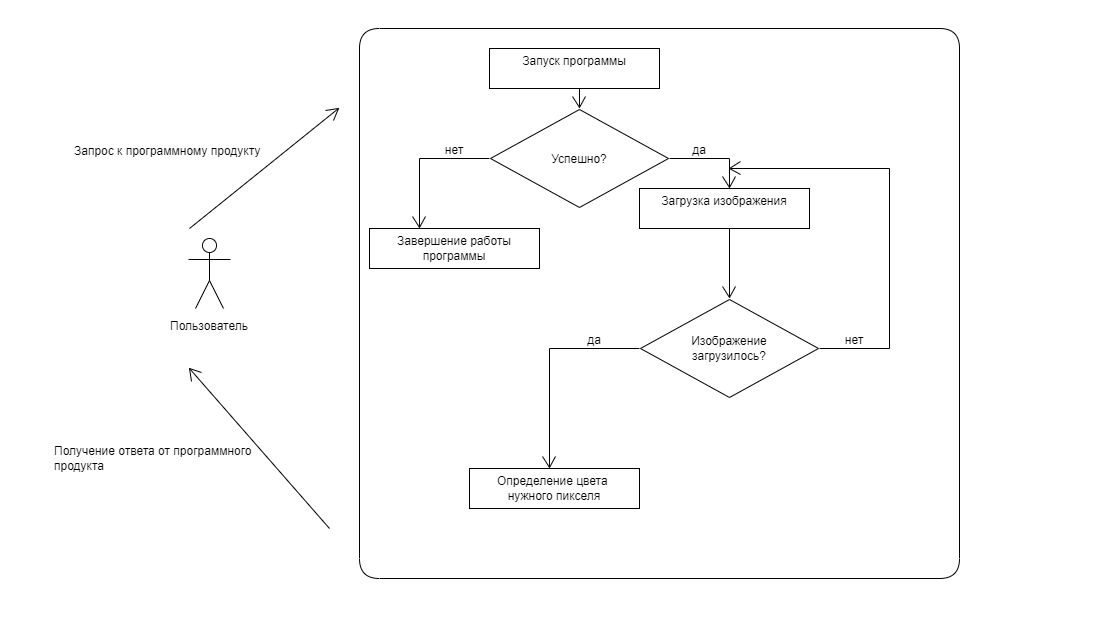
**Недостатки:**

1. Отсутствует возможность загрузки фотографии и определения пикселя по ней

### Используемые технические и программные средства

При написании приложения использовалась IDE QtCreator. Само же приложение написано на языке c++ с использованием библиотеки “QtWidgets”. Также стоит упомянуть, что при написании и тестировке программы использовалась ОС windows 10.

### Формализация бизнес-процессов предметной области

****

### Описание программы

Программа PixelPicker предназначена для определения цвета пикселя в изображении. Чтобы определить цвет пикселя достаточно загрузить изображение и навестись мышкой на нужный пиксель и тогда программа определяет цвет пикселя в 2 форматах:

1. В графическом
2. В формате RGB

### Исходный код

**#pixelpicker.pro**

QT+=widgets

SOURCES+=main.cpp

**//main.cpp**

#include <QtWidgets>

QColor color;

QString filePath="";

class Window : public QWidget{

public:

Window();

void openButtonClick();

void openImage (const QString &path=QString());

private:

QLabel \*textLabel;

QLabel \*label;

QLabel \*labelRGB;

QLabel \*labelR;

void *mouseMoveEvent*(QMouseEvent \*event);

QLineEdit \*pathEdit;

QPushButton \*openButton;

void *paintEvent*(QPaintEvent \*event);

QPixmap pixmap1;

};

Window::Window(){

resize(1500,900);

setWindowTitle("PixelPicker");

label = new QLabel(this);

label->setGeometry(0,0,200,300);

label->setFont(QFont("Arial", 10));

textLabel=new QLabel(filePath, this);

textLabel->setGeometry(0,0,250,20);

labelR=new QLabel(this);

labelR->setGeometry(0,425,200,200);

labelR->setText("R=");

labelR->setFont(QFont("Arial", 20));

labelRGB=new QLabel(this);

labelRGB->setGeometry(40,425,250,200);

labelRGB->setFont(QFont("Arial", 20));

setMouseTracking(true);

pathEdit=new QLineEdit(this);

pathEdit->setGeometry(0,30,250,20);

pathEdit->setText(filePath);

openButton=new QPushButton("Open photo", this);

openButton->setGeometry(0,60,80,30);

connect(openButton, &QPushButton::clicked, this, &Window::openButtonClick);

}

void Window::*mouseMoveEvent*(QMouseEvent \*event){

int x = event -> x();

int y = event -> y();

label->setText(QString("Mouse coordinates: [%1;%2]").arg(x).arg(y));

QWidget::grab(QRect(x, y, 1, 1)).toImage().pixelColor(0, 0);

qDebug()<<event->globalX()<<event->globalY()<<endl;

QScreen \*screen=QGuiApplication::primaryScreen();

if (!screen){

return;

}

QPixmap pixmap=screen->grabWindow(0);

QImage img=pixmap.toImage();

QRgb rgb=img.pixel(event->globalX(), event->globalY());

color.setRgb(rgb);

labelRGB->setText(QString::number(color.red())+" G=" + QString::number(color.green()) + " B=" + QString::number(color.blue()));

}

void Window::openButtonClick(){

filePath=pathEdit->text();

openImage();

update();

}

void Window::openImage(const QString &path){

filePath=path;

if (filePath.isNull()){

filePath=QFileDialog::getOpenFileName(this, "Open Image", " ", "Image Files(\*.png \*.jpg \*.bmp)");

}

if (!filePath.isEmpty()){

if (!pixmap1.load(filePath)){

QMessageBox::warning(this, "Open Image", "The Image file could not be loaded", QMessageBox::Cancel);

return;

}

}

textLabel->setText(filePath);

pathEdit->setText(filePath);

}

void Window::*paintEvent*(QPaintEvent \*){

QPainter painter1(this);

QPainter painter2(this);

pixmap1.load(filePath);

QPixmap scaledPixmap = pixmap1.scaled(900,1024, Qt::KeepAspectRatio, Qt::SmoothTransformation);

painter1.drawPixmap(600,0, scaledPixmap);

painter2.setBrush(color);

painter2.drawRect(50,400,50,50);

update();

}

int main(int argc, char \*argv[])

{

QApplication a(*argc*, argv);

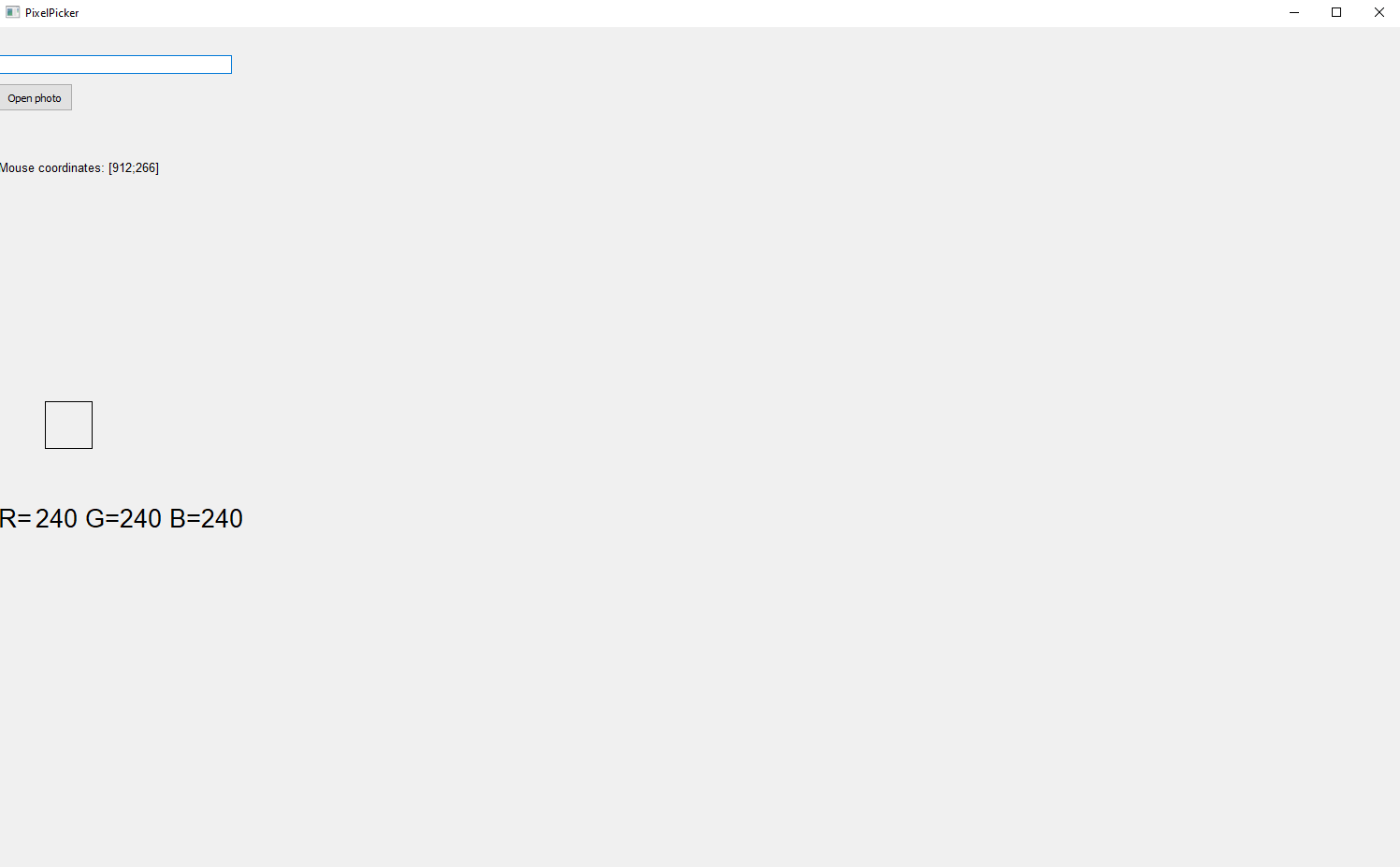
Window w;

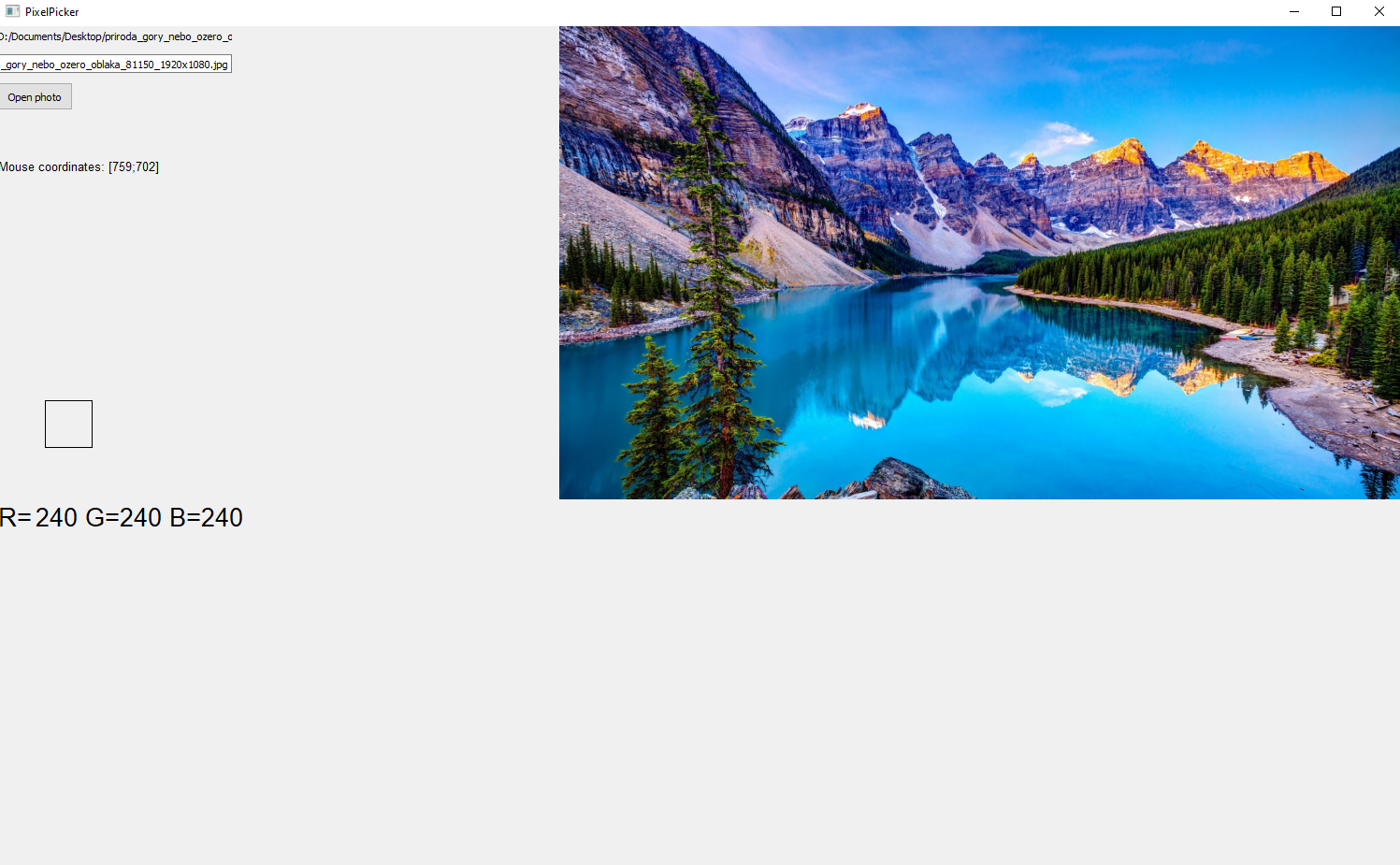
w.show();

return a.exec();

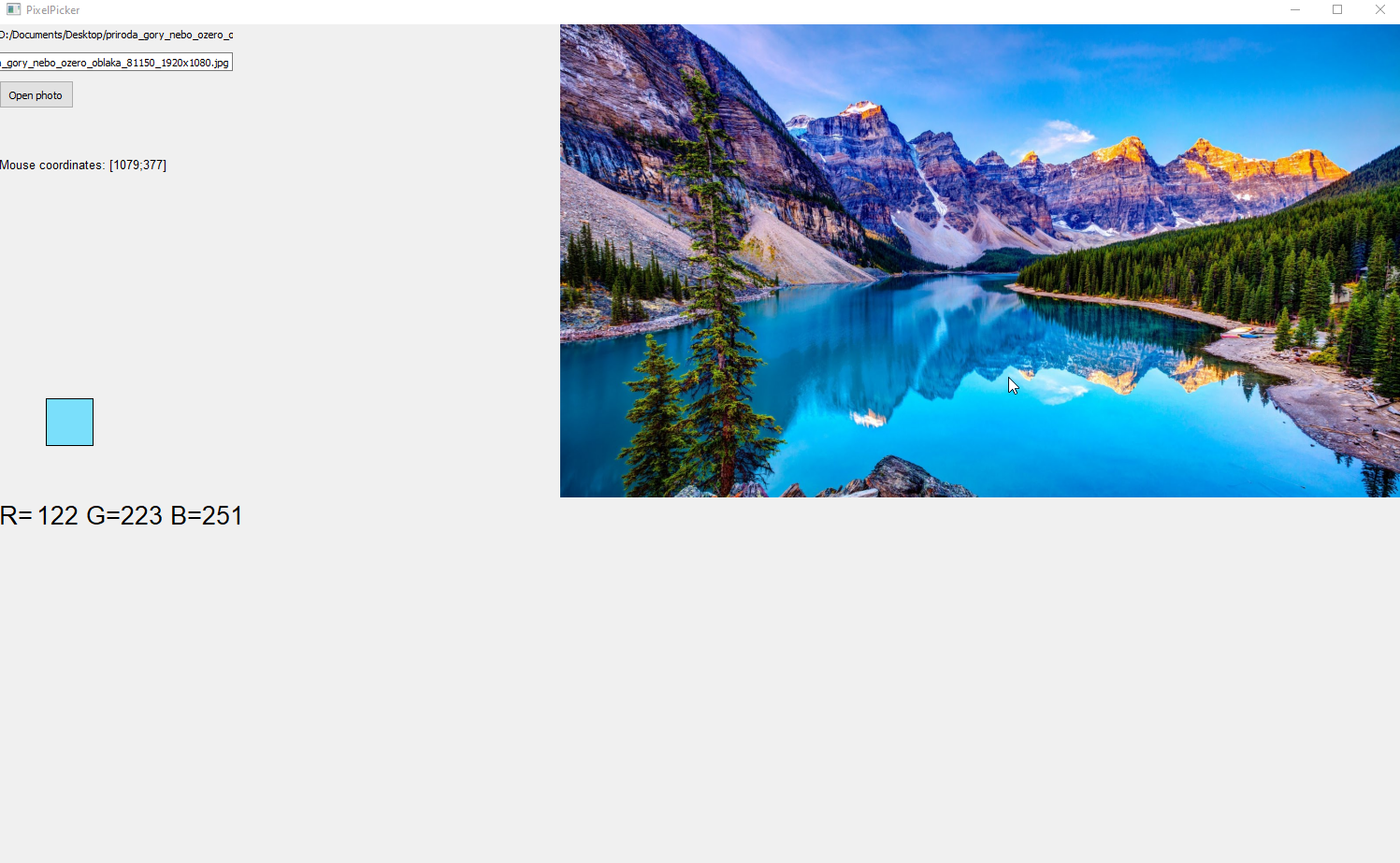
}

### Отчет по тестированию

1. При тестировании было проверено исчисление программой координат курсора мыши. Тестирование прошло успешно
2. Была проверена возможность загрузки изображений. Проверка прошла успешно.



1. Было проведено тестирование на правильность определения оттенков/цветов. Тестирование прошло успешно.



### Руководство пользователя

1. Откройте приложение используя ярлык на рабочем столе
2. Используя кнопку “Open photo” загрузите нужное вам изображение
3. Курсором мыши водите по изображению
4. Найдите нужный вам пиксель и определите его цвет

### Руководство системного администратора

1. Скачайте и установите программный продукт «PixelPicker»
2. Создайте ярлык на рабочем столе
3. Убедитесь в правильности работы программного продукта

### Список источников

1. Trello. URL: <https://trello.com/>
2. Проект на github. URL: <https://github.com/RareSkye/PixelPicker>
3. Google Drive. URL: <https://drive.google.com/>
4. Ecolor. URL: <http://www.evileg.ru/ecolor>
5. Color picker. URL: <https://www.google.com/search?q=Color+picker&oq=Color+picker&aqs=chrome..69i57j69i59j0l5j69i60.347j0j7&sourceid=chrome&ie=UTF-8>
6. colourco. URL: <https://colourco.de/>
7. UMLetino. URL: <http://www.umlet.com/umletino/umletino.html>